

# 并行知识库机的知识划分及其推理

廖明宏 郭福顺 程退安

(哈尔滨工业大学计算机系, 150001)

**摘 要** 并行知识库机是一种采用并行处理技术来提高大规模知识库系统性能的专用机。其中, 知识的划分及并行推理模型是两个关键问题。本文以产生式系统作为并行知识库机的基本知识表示方法, 重点讨论产生式系统的划分和并行推理问题。

**关键词** 并行知识库机, 产生式系统, 划分, 并行推理

## 1 引 言

知识库系统与知识库机的概念是日本在五代机计划中首先使用的。但对五代机(日本将其归结为当代计算机科学技术中两项最重要技术的结合: 知识处理和大规模并行处理技术)的研究, 日美的看法略有不同。日本的特点是理论驱动式的研究, 提出了一套理论, 制订了五代机计划。美国的特点是应用驱动式的研究, 注重于在通用高级并行体系结构上, 用 LISP 语言, 产生式系统研究 AI 问题的求解。

知识处理机至今没达到预期的进展和应用水平, 但还是有相当的进步, 为将来发展积累了扎实、宝贵的经验。特别是日本对五代机成果评价很高(虽然也承认五代机计划及其实施都有问题), 认为他们研制的五代机原型系统是目前世界上最快的、规模最大的知识处理系统。因此他们进而考虑五代机技术面对市场的问题, 由国际工商部门于 1993 年 7 月制订了五代机系统延续计划(FGCS Follow-on project)<sup>[1]</sup>。

我们的研究工作特点接近美国方式; 考虑到国情, 根据实际应用的需要, 在知识库机硬件体系结构的研制上采用较成熟的技术, 但在这种体系结构的支持下着重研究产生式系统的划分和并行推理问题。

## 2 产生式系统语言: OPS<sub>5</sub>

一个产生式系统是由一个规则库, 一个数据库和一个推理机三部分组成。其中, 规则库存放具有“IF...THEN”形式的规则知识; 数据库存放推理中的事实数据; 推理机用规则库和数据库进行推理。

OPS<sub>5</sub> 语言是产生式系统语言的代表。在 OPS<sub>5</sub> 中规则库称为产生式存储器(PM), 它用

收稿日期: 1994-04-11。该项研究为国家自然科学基金资助项目。廖明宏, 讲师、博士、研究方向为并行知识库机。  
郭福顺, 教授, 研究方向为并行处理、人工智能、操作系统。程退安, 教授, 研究方向为智能机体系结构、智能网络。

于存放一组产生式规则。规则的形式为：

$$(P \langle \text{规则名} \rangle \langle \text{LHS} \rangle \rightarrow \langle \text{RHS} \rangle)$$

其中：〈LHS〉是规则的左手部。表示条件；

〈RHS〉是规则的右手部，表示结论。

数据库在 OPS<sub>5</sub> 中称为工作存储器(WM)，用以存放在推理过程中产生的当前数据元素：工作存储元素(wme)，wme 的形式为：

$$(\langle \text{类名} \rangle \uparrow \langle \text{属性} \rangle \langle \text{属性值} \rangle \dots \uparrow \langle \text{属性} \rangle \langle \text{属性值} \rangle)$$

OPS<sub>5</sub> 的推理机控制整个产生式系统的执行过程，它由匹配——冲突归结——点火三个阶段的循环组成。

阶段 1—匹配：把产生式每个条件元素与 WM 中每个元素进行比较，看哪个可与条件元素匹配。如果至少有一个 wme 与之匹配，则该条件被满足；如果产生式 LHS 所有的条件都被满足，则称该产生式被满足。匹配过程的输出是一个冲突集，冲突集内的对象叫做示例。

阶段 2—冲突归结：根据一定的策略从冲突集中选择一个示例，而这个示例将在点火阶段被执行。若选择不出一个示例。则推理过程结束。

阶段 3—点火：选中的产生式的 RHS 动作被依次执行。重复执行上述三个阶段的过程。

在这个推理过程中，匹配阶段最费时间。为此，C. Forgy 提出一种有效的匹配算法：Rete 算法<sup>(2)</sup>。它的基本思想就是根据所有产生式规则左手部条件，构造一个数据驱动的识别网络(称为 Rete 网)，匹配时将每个 wme 从 Rete 网的根结点出发向下流动匹配，网的叶结点是匹配成功的冲突集。由于 Rete 网采取存储中间匹配结果和共享匹配操作等方法，提高了匹配阶段的运行速度。因此 Rete 算法成为产生式系统的基本匹配算法。

Rete 算法并不能根本解决产生式系统推理效率低的问题。为此又出现了许多并行匹配算法和其它并行处理技术<sup>(3)</sup>。在这些并行处理方法中，多规则点火系统(Multiple Rule Firing System)<sup>(4)</sup>是一种较有前途的做法。它的基本思想是在产生式系统的一个推理周期中，尽可能

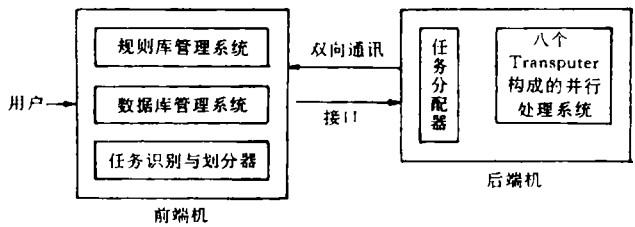


图 1 并行知识库机的硬件结构

多地点火多条规则，从而使整个推理周期的三个阶段都并行处理，以此来提高系统的推理效率。但为保证多规则点火系统与顺序点火系统结果的等价性，必须解决相容性(Compatibility)和汇流性(Convergence)问题。所谓相容性就是要求并行点火的规则必须是不相互矛盾的。而汇流性问题是相容性条件的补充，它要求从全局上保证多规则点火系统的正确性。目前许多实验系统对这两个问题的解决方法各不相同，对系统推理效率的提高也有所不同，但到目前为止，还没找出一种公认的有效办法。<sup>(4)</sup>

本文旨在一种由前后端机组成的并行知识库机硬件环境中设计产生式系统的多规则点火系统，以期得到较好的推理效率。

3 并行知识库机的硬件结构

并行知识库机的硬件结构如图 1 所示。

从硬件结构看，并行知识库机的特点是：

- (1) 由一台 486 微机组成前端机，为规则库管理系统，数据库管理系统和任务识别与划分器提供活动场所。
- (2) 后端机是由 8 个 Transputer 互连而成的并行处理系统。Transputer 是一种采用 VL-SI 技术，把一台完整的计算机集成到一块芯片上的芯片型计算机，它基于 RISC 体系结构。多个 Transputer 易于互连成不同的拓扑结构，构成各种并行处理系统。它提供的 occam 语言能直接支持程序的并行和同步<sup>[5]</sup>。从宏观上看，后端机口是前端机的一块插板。
- (3) 前后端机由自行设计的高速双向通讯接口电路连接。

4 产生式系统的划分

为使产生式系统能在图 1 给出的多机环境中运行，有必要对一个产生式系统进行划分。具体地，把一个产生式系统分解成若干个相对独立的子系统，并分配到不同的处理机上。每个子系统由若干规则子模块及其相应的数据库组成。每台处理机对各自的子系统进行推理。所有的处理机用异步的工作方式运行。它们之间采用消息传递方式进行通讯，以此来协调同一个产生式系统。

系统的划分准则如下：

- (1) 相关程度大的规则尽量放在同一子系统中，以减少处理机之间的通讯开销；
- (2) 可并行规则尽可能放在不同的子系统中，以提高处理机的利用率；
- (3) 各子系统的工作量应尽可能均匀，以达到系统的负载平衡。

4.1 工作存储器划分成若干个关系数据库

OPS<sub>5</sub> 的工作存储器与关系数据库存在一种自然对应关系，见表 1。

由表 1 可以看出，OPS<sub>5</sub> 中的一个类对应关系数据库中的一个关系。因此，可以为每一个类建立一个关系数据库，数据库的结构由类的属性域来确定。把工作存储器元素按它们的类划分成不同的数据库，使得在匹配时，每条规则只关心与之有关的数据库记录而不必理会其它数据库上的记录。

表 1 工作存储器与关系数据库的对应关系

工作存储器	关系数据库
类	关系
属 性	域
工作存储元素	记录

4.2 规则库按规则相关程度划分成若干规则子模块

规则库的划分就是根据系统的划分准则将规则库划分成若干相对独立的子模块。规则库的划分算法很多，这里介绍我们提出的一种更为有效的划分算法：按规则相关度划分算法：DDPA。

假设，RB：表示规则库； $R_i, R_j \in RB$ ；MSET：表示模块集； $M_i, M_j \in MSET$ ； $R_0$ ：表示大于等于 0 的实数集。  
则有如下定义：

〔定义 1〕规则间的相关函数：对给定的规则库 RB，可定义一个函数  $f$ ，使得对 RB 中任意两条规则  $R_i$  和  $R_j$ ，有  $f(R_i, R_j) \in R_0$ ，则称函数  $f$  为规则间的相关函数。

〔定义 2〕模块间的相关函数：对一给定的模块集 MSET，可定义一个函数  $F$ ，使得对 MSET 中任意两个模块  $M_i, M_j$  有  $F(M_i, M_j) \in R$ 。则称函数  $F$  为模块间的相关函数。

相关函数的赋值根据以下几条原则：

(1) 冲突规则间的相关函数赋给一个最大值；

(2) 非冲突规则间的相关函数为

$$f(R_i, R_j) = |(LD_i \cap RD_j) \cup (LD_j \cap RD_i)|$$

其中： $LD_i$  表示  $R_i$  的左手部引用的数据库集合； $RD_i$  表示  $R_i$  的右手部引用的数据库集合。符号  $||$  表示求集合元素个数。

(3) 模块间相关函数  $F(M_i, M_j)$  的赋值。

$$F \text{ 的整数部分} = |MD_i \cup MD_j|$$

$$F \text{ 的小数部分} = |MN_i - MN_j|$$

其中： $MD_i$  为模块  $M_i$  的引用数据库集：

$$MD_i = \bigcup_{k=1}^m (LD_{ik} \cup RD_{ik}) \quad (\text{设 } m \text{ 为 } M_i \text{ 的规则数})。$$

$MN_i$  表示模块  $M_i$  的规则数。

相关函数可以用来刻画规则(或模块)之间的相关程度(简称相关度)。函数值越大，表示规则(或模块)之间的相关程度也越大。

对一给定的规则库进行划分，其划分结果不唯一。一般可根据系统中的处理机台数，给出欲划分的模块数。DDPA 算法的基本思想是：

$S_1$ 〔初始化〕：把规则库中每条规则看成一个模块；

$S_2$ 〔按冲突规则合并〕：把每对相互冲突的规则合并成同一模块；

$S_3$ 〔按规则相关度合并〕：若  $S_2$  没达到给定模块数的要求，则对其它规则按其规则相关度的大小由大到小进行两两模块合并；

$S_4$ 〔按模块相关度合并〕：若  $S_3$  没达到给定模块数的要求，则按模块间的相关度大小由大到小进行两两模块合并，直到满足给定模块数的要求。

## 5 产生式系统的并行推理

为使产生式系统在多机环境中进行并行推理，我们提出一种新的并行推理模型：目标驱动的并行推理模型 GPIM。它是一种正反向推理相结合的多规则点火系统。GPIM 的基本思想是：

1) 在多机系统中，指定任意一台处理机为主机；其它处理机为从机。初始时，规则库和数据库存放在主机上；

2) 主机把规则库中的子模块及其相关的数据库分布到其它处理机上；

3) 主机把规则库中给出的推理目标与规则库中所有规则的右手部进行比较，找到与目标相匹配的所有规则。把这些规则的左手部作为新的子目标，以广播方式发给其它处理机(这一工作类似反向推理)。

4) 从机收到与之有关的子目标, 开始进行正向推理。其推理算法采用 Rete 算法。

5) 从机之间采用异步工作方式。所有的处理机同时进行各自的推理工作。并采用消息传递方式进行通讯。即当某一处理机点火一条规则并产生对公共数据库(被两个以上模块所引用的数据库)的操作(插入、删除或修改等)时, 它首先修改本机的数据库, 然后把修改信息传送到引用该数据库的所有处理机上。接收端上的处理机暂时中断推理工作, 接收发来的消息, 并做相应的修改, 然后继续推理。

6) 当某一从机推导出一个子目标, 它将该目标返回给主机, 然后继续它的正向推理工作。

7) 当某一从机推理结束, 它返回给主机一个等待信息。然后等待其它处理机发给它新的对数据库操作的信息, 或由主机发给它的控制信息。若其它处理机促使它继续工作, 它首先返回给主机一个忙信息。然后继续推理。

8) 主机收到从机发来的子目标时, 将它与目标规则的左手部进行匹配。若目标规则满足, 则点火该目标规则。然后通知从机, 终止各自的推理工作, 最后整个系统推理结束。若目标规则没被满足, 主机继续等待从机返回的子目标。

9) 若所有从机都返回给主机等待信息, 且此时目标规则还没被满足, 主机返回给用户目标没被满足的信息。然后拷贝回从机上最新的规则库和数据库, 并结束整个推理过程。

从以上 GPIM 的基本思想可以看出, 它能够产生式系统的各个子系统, 分布到不同的处理机上, 并协调各处理机进行并行推理, 以达到对整个系统推理的目的。

GPIM 模型可以简单地解决相容性和汇流性问题。其中, 相容性问题可以在规则划分过程中得到解决。因为规则划分算法首先将相互冲突的规则划分到同一模块, 它们被分布到同一处理机上, 从而避免相互冲突规则同时点火的现象。汇流性问题的解决很简单, 它只要求用户给出推理目标, GPIM 就可以根据推理目标引导整个推理过程沿着推理目标方向进行, 从而保证推理结果的正确性。

## 6 模拟结果

根据上述并行推理模型, 我们对四个实例进行了模拟实验, 这四个实例是:

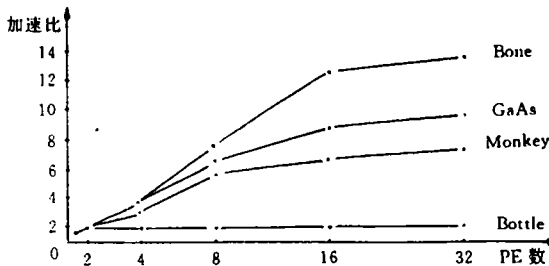


图 2 模拟结果

- 1) Bottle: 即水罐问题, 规则数 6
- 2) Monkey: 猴子香蕉问题, 规则数 20
- 3) GaAs: 砷化钾生产过程控制专家系统, 规则数 64
- 4) Bone: 骨肿瘤诊断专家系统, 规则数 302

实验结果如图 2 所示

由图 2 可见, 该并行推理模型适合大规模产生式系统的推理。

## 7 结束语

并行知识库机的研究是一个前沿课题。我们在硬件结构设计上尽量采用成熟的设备和技术, 而把重点放在并行自动推理的研究。经过对几个专家系统的模拟实验, 结果表明该处理方法可显著提高产生式系统的推理效率。

### 参 考 文 献

- [ 1 ] Institute for New Generation Computer Technology(ICOT), Outline of ICOT and FGCS Follow—on Project, 1993, 7.
- [ 2 ] C. L. Forgy. Rete: A Fast Algorithm for the Many Pattern/Many Object Pattern Match Problem, Artificial Intelligence 19, Sept, 1982.
- [ 3 ] Anoop Gupta. Parallelism in Production Systems, Morgan Kaufmann Publishers, Inc, California, 1987.
- [ 4 ] Steve Kuo and Dan Moldovan. The State of Art in Parallel Production Systems, Journal of Parallel and Distributed Computing 15, Sept, 1992.
- [ 5 ] David May. The Next Generation Transputers and Beyond, Distributed Memory Computing, 2nd European Conf. EDMCC2, 1991.

## KNOWLEDGE PARTITION AND INFERENCE OF THE PARALLEL KNOWLEDGE BASE MACHINE

Liao Minghong Guo Fushun Cheng Tuian

(Dept. of Computer Science, Harbin Institute of Technology, 150001)

**Abstract** The parallel knowledge base machine is a special computer to improve the performance of the large knowledge base system by using of the technique of parallel processing, The knowledge partition and parallel inference model are its two important problems. We select the production systems as the fundamental knowledge representation of the parallel knowledge base machine, the paper focus on the partition and parallel inference problems of the production systems.

**Key words** Parallel knowledge base machine, Production systems, Partition, Parallel inference